|  |  |  |  |
| --- | --- | --- | --- |
| **Object Oriented Programming Lab 07** | | | |
| **Course Code:** | COMP-112L | **Class** | CS (B) |
| **Lab Engineer** | Laiba Khalid | **Semester** | 2nd |
| **Lab Title** | C++ Classes | **Section** | - |
| **Name** |  | **Reg no.** |  |
| **Content Covered** | C++ Class Implementation. | | |
| **Instructions:**  • Submit the file with your names following your registration numbers like AI001\_Name.  • Submit soft copy of the report before deadline. Marks will be deducted for late submissions. | | | |

**Inheritance in C++(single Inheritance, Multiple Inheritance & Multi-level inheritance)**

**Introduction:**

**Inheritance** is a fundamental concept in Object-Oriented Programming (OOP) that allows a class (child/derived class) to acquire properties and behaviors of another class (parent/base class).

* **Sub Class (Child Class)**: The class that inherits properties from another class is called Sub class or Derived Class.
* **Super Class (Parent Class)**: The class whose properties are inherited by a sub-class is called Base Class or Superclass.

**Example**: Dog, Cat, Cow can be Derived Class of Animal Base Class.

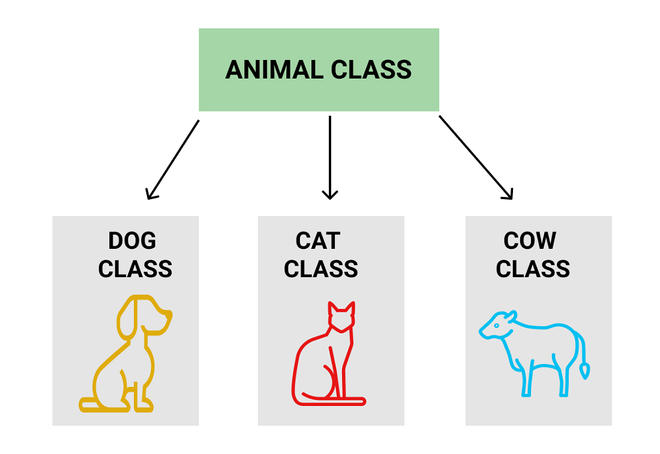


Figure 1: Inheritance in C++

**Syntax:**

|  |
| --- |
| class ParentClass {  // Parent class members (attributes & methods)  };  class ChildClass : public ParentClass { // Child class inherits from ParentClass  // Child class members  }; |

**Types of Inheritance in C++:**

1. **Single Inheritance** - One derived class inherits from a single base class.
2. **Multiple Inheritance** - A derived class inherits from more than one base class.
3. **Multilevel Inheritance** - A derived class inherits from another derived class.

**Single Inheritance:**

**Example:**

|  |
| --- |
| #include <iostream>  using namespace std;  class Vehicle { // Base class  public:  void fuelType() {  cout << "Uses fuel" << endl;  }  };  class Car : public Vehicle { // Derived class  public:  void start() {  cout << "Car starts with a key" << endl;  }  };  int main() {  Car myCar;  myCar.start(); // Output: Car starts with a key  myCar.fuelType(); // Output: Uses fuel (inherited)  return 0;  } |

**Practice Task:**

1. Create a base class Person with name and age attributes. Create a derived class Student that adds a grade attribute. Implement a function to display details.

**Multiple Inheritance:**

**Example:**

|  |
| --- |
| #include <iostream>  using namespace std;  class Engine {  public:  void power() {  cout << "Engine is running" << endl;  }  };  class Wheels {  public:  void rotate() {  cout << "Wheels are rotating" << endl;  }  };  class Car : public Engine, public Wheels {  public:  void drive() {  cout << "Car is driving" << endl;  }  };  int main() {  Car myCar;  myCar.power(); // From Engine class  myCar.rotate(); // From Wheels class  myCar.drive(); // From Car class  return 0;  } |

**Practice Task:**

1. Create a Device class with brand. Create a **TouchScreen** class with **touchType**. Create **Smartphone** that inherits both.

**Multilevel Inheritance:**

**Example:**

|  |
| --- |
| #include <iostream>  using namespace std;  class Animal {  public:  void eat() {  cout << "Eating..." << endl;  }  };  class Mammal : public Animal {  public:  void breathe() {  cout << "Breathing..." << endl;  }  };  class Dog : public Mammal {  public:  void bark() {  cout << "Barking..." << endl;  }  };  int main() {  Dog myDog;  myDog.eat(); // Inherited from Animal  myDog.breathe(); // Inherited from Mammal  myDog.bark(); // Defined in Dog  return 0;  } |

**Practice Task:**

1. Create a **Shape** class with a draw() method. Create **Rectangle** derived from Shape with length and width attributes. Create **ColoredRectangle** derived from Rectangle that adds color.